# Approximating $k$-Median via Pseudo-Approximation

Shi Li
Princeton, USA
shili@cs.princeton.edu

Ola Svensson
EPFL, Switzerland
ola.svensson@epfl.ch

November 2, 2012



**Abstract**

We present a novel approximation algorithm for $k$-median that achieves an approximation guarantee of $1 + \sqrt{3} + \epsilon$, improving upon the decade-old ratio of $3 + \epsilon$. Our approach is based on two components, each of which, we believe, is of independent interest.

First, we show that in order to give an $\alpha$-approximation algorithm for $k$-median, it is sufficient to give a *pseudo-approximation algorithm* that finds an $\alpha$-approximate solution by opening $k + O(1)$ facilities. This is a rather surprising result as there exist instances for which opening $k + 1$ facilities may lead to a significant smaller cost than if only $k$ facilities were opened.

Second, we give such a pseudo-approximation algorithm with $\alpha = 1 + \sqrt{3} + \epsilon$. Prior to our work, it was not even known whether opening $k + o(k)$ facilities would help improve the approximation ratio.

## 1 Introduction

Suppose you wish to select $k$ polling stations for the US election so as to minimize the average distance each voter has to travel to his/her closest polling station. Then you need to solve the classic NP-hard $k$-median problem that we shall design better approximation algorithms for in this paper. Formally, a $k$-median instance $\mathcal{I}$ is defined by the tuple $(k, \mathcal{F}, \mathcal{C}, d)$, where $k$ is the number of facilities allowed to be opened, $\mathcal{F}$ is a set of potential facility locations, $\mathcal{C}$ is a set of clients, and $d$ is a distance metric over $\mathcal{F} \cup \mathcal{C}$. The goal is to open a set $\mathcal{S} \subseteq \mathcal{F}$ of $k$ facilities so as to minimize $cost_{\mathcal{I}}(\mathcal{S}) = \sum_{j \in \mathcal{C}} d(j, \mathcal{S})$, where $d(j, \mathcal{S})$ denotes the distance from $j$ to its nearest facility in $\mathcal{S}$. When $\mathcal{F} = \mathcal{C} = X$, a solution $\mathcal{S}$ partitions the set of points into what is known as clusters and thus the objective measures how well $X$ can be partitioned into $k$ clusters. The $k$-median problem has numerous applications, starting from clustering and data mining [3] to assigning efficient sources of supplies to minimize the transportation cost([14, 20]).

The difficulty of the $k$-median problem lies in the hard constraint that only $k$ facilities are allowed to be opened. Indeed, without such a constraint, we could simply open all facilities. Early approaches [18, 17, 13] overcame this difficulty by giving pseudo-approximations that obtain better guarantees while violating the mentioned constraint by opening $k + \Omega(k)$ facilities. The first constant factor approximation algorithm that opens $k$ facilities is due to Charikar et al. [6]. Based on LP rounding, their algorithm produces a $6\frac{2}{3}$-approximation. Several of the ideas in [6] are inspired from constant factor approximation algorithms obtained for the closely related metric uncapacitated facility location (UFL) problem. The UFL problem has similar input as $k$-median but instead of giving an upper bound $k$ on the number of facilities we can open, it specifies an

opening cost $f_i$ for each facility $i \in \mathcal{F}$. The goal is to open a set of facilities $\mathcal{S}$ that minimizes the sum of the opening costs and connection costs, i.e., $\sum_{i \in \mathcal{S}} f_i + \mathsf{cost}_{\mathcal{I}}(\mathcal{S})$.

The connection between UFL and $k$-median is motivated by basic economic theory: if we let the opening costs of facilities be small then a "good" solution to UFL will open many facilities whereas if we let the opening costs of facilities be large then a good solution will only open few facilities. By appropriately selecting the cost of facilities, one can therefore expect that an algorithm for UFL opens close to $k$ facilities and therefore almost also gives a solution to the $k$-median problem. This is the intuition of the concept of bi-point solutions that we define in Section 1.2. Jain and Vazirani first exploited this concept in a beautiful paper [12] to obtain a 6-approximation algorithm for $k$-median using their 3-approximation primal-dual algorithm for UFL. The factor 3 was later improved by Jain et al. [11] to 2 resulting in a 4-approximation algorithm for $k$-median.

In spite of the apparent similarities between UFL and $k$-median, current techniques give a considerable better understanding of the approximability of UFL. For UFL and its variants, there has indeed been a steady stream of papers giving improved algorithms [17, 22, 12, 7, 13, 5, 10, 11, 19, 4]. The current best approximation algorithm is due to Li [16]. He combined an algorithm by Byrka [4] and an algorithm by Jain et al [11] to achieve an approximation guarantee of 1.488. This is close to being best possible, as it is hard to approximate UFL within a factor of 1.463 [9]. In contrast there has been less progress for $k$-median and the approximability gap is larger. The best known approximation algorithm is the local search algorithm given by Arya et al. [2]. They showed that if there is a solution $\mathcal{F}'$, where any $p$ swaps of the open facilities cannot improve the solution, then $\mathcal{F}'$ is a $3 + 2/p$ approximation. This leads to a $3 + \epsilon$ approximation that runs in time $n^{2/\epsilon}$. On the negative side, Jain et al. [11] proved that the $k$-median problem is hard to approximate within a factor $1 + 2/e \approx 1.736$. Moreover, the natural linear programming relaxation of $k$-median is known to have an integrality gap of at least 2. The best upper bound is by Archer et al. [1], who showed that the integrality gap is at most 3 by giving an exponential time rounding algorithm that requires to solve the maximum independent set problem.

As alluded to above, the main difficulty of the $k$-median problem is the hard constraint that we can open at most $k$ facilities. In this paper we take a different approach that allows us to relax this constraint and thereby addressing the problem from a novel point of view using what we call a pseudo-approximation algorithm. This leads to the improved approximation algorithm breaking the barrier of 3 that we discuss next.

### 1.1 Our Results

Our improved approximation algorithm can be stated as follows.

**Theorem 1.** *There is an algorithm which, given a $k$-median instance $\mathcal{I}$ and a number $\epsilon > 0$, produces a $1 + \sqrt{3} + \epsilon$-approximate solution to $\mathcal{I}$ in running time $O\left(n^{O(1/\epsilon^2)}\right)$.*

Our algorithm contains two main components, each of which, we believe, is of independent interest. First, we show that in order to give an approximation algorithm for $k$-median, it suffices to give a *pseudo-approximation algorithm* $\mathcal{A}$ which, given a $k$-median instance $\mathcal{I}$, outputs a set $\mathcal{S} \subseteq \mathcal{F}$ of $k + c$ facilities with $\mathsf{cost}_{\mathcal{I}}(\mathcal{S}) \leq \alpha \mathsf{opt}_{\mathcal{I}}$, where $\mathsf{opt}_{\mathcal{I}}$ is the cost of optimum solution for $\mathcal{I}$. Given such an algorithm $\mathcal{A}$ as a black box, we can design an $\alpha + \epsilon$-approximation algorithm $\mathcal{A}'$ whose running time is $n^{O(c/\epsilon)}$ times that of $\mathcal{A}$. Interestingly, the instance (see Figure 1) that gives the integrality gap of 2 for the natural LP relaxation of $k$-median vanishes if we allow the integral

solution to open $k+1$ facilities. This suggests that our reduction may bring in new avenues for approximating $k$-median. In particular, we find the following open problem interesting: given a $k$-median instance $\mathcal{I}$, what is the maximum ratio between the cost of the optimum integral solution of $\mathcal{I}$ with $k+1$ open facilities, and the LP value (with $k$ open facilities)?

To complement the first component, we give the aforementioned pseudo-approximation algorithm $\mathcal{A}$ with $\alpha = 1+\sqrt{3}+\epsilon$. Prior to our work, it was not even known whether opening $k+o(k)$ facilities would help improve the approximation ratio; all known pseudo-approximation algorithms require $k+\Omega(k)$ open facilities. In contrast, our algorithm only opens $k+O(1/\epsilon)$ facilities. The algorithm $\mathcal{A}$ contains 2 steps. We obtain a *bi-point solution* for $k$-median using the algorithm of [11]. We lose a factor of 2 in this step. Then, we convert the bi-point solution into an integral solution with $k+O(1/\epsilon)$ open facilities, losing another factor of $\frac{1+\sqrt{3}+\epsilon}{2}$ in the approximation ratio. We remark that if we had insisted on opening $k$ facilities, then a factor of 2 has to be lost in the last step as the instance achieving an integrality gap of 2 has a bi-point solution.

Theorem 1 does not give a better upper bound on the integrality gap of the natural LP due to the following reason: instead of running the pseudo-approximation algorithm $\mathcal{A}$ on the input instance $\mathcal{I}$, we run it on a residual instance $\mathcal{I}'$ obtained from $\mathcal{I}$ by removing a subset of facilities that the optimal solution does not open. The way we obtain $\mathcal{I}'$ is to guess $O(1/\epsilon^2)$ "events" and let $\mathcal{I}'$ be the instance conditioned on these events. Due to this nature, our algorithm can be converted to a rounding algorithm based on solving an $O(1/\epsilon^2)$-level LP in the Sherali-Adams hierarchy. Instead of guessing the $O(1/\epsilon^2)$ events, we can now find these events explicitly by looking at the LP solution. Conditioning on these events, we obtain a fractional solution of the basic LP. By rounding this LP, we obtain a $1+\sqrt{3}+\epsilon$-approximate solution. Thus, our approach can be seen to give an $1+\sqrt{3}+\epsilon$-upper bound on the integrality gap of the $O(1/\epsilon^2)$-level LP in the Sherali-Adams hierarchy. Our result was in fact first obtained by studying the power of the Sherali-Adams hierarchy for the $k$-median problem. However, as it can also be obtained using a combinatorial approach with less cumbersome notation, we have chosen to present that approach.

### 1.2 Preliminaries

Given a $k$-median instance $\mathcal{I} = (k, \mathcal{F}, \mathcal{C}, d)$, a *pseudo-solution* to $\mathcal{I}$ is a set $\mathcal{S} \subseteq \mathcal{F}$. A pseudo-solution $\mathcal{S}$ satisfying $|\mathcal{S}| \le k$ is a *solution* to $\mathcal{I}$; a pseudo-solution $\mathcal{S}$ with $|\mathcal{S}| \le k+c$, for some number $c \ge 0$, is called a *c-additive (pseudo-)solution*. The cost of a pseudo-solution $\mathcal{S}$ to $\mathcal{I}$ is defined as $\mathsf{cost}_{\mathcal{I}}(\mathcal{S}) = \sum_{j \in \mathcal{C}} d(j, \mathcal{S})$, where $d(j, \mathcal{S})$ denotes the distance from $j$ to its closest facility in $\mathcal{S}$. We let $\mathsf{OPT}_{\mathcal{I}}$ denote an optimal solution to $\mathcal{I}$, i.e., one of minimum cost, and we let $\mathsf{opt}_{\mathcal{I}} = \mathsf{cost}_{\mathcal{I}}(\mathsf{OPT}_{\mathcal{I}})$. To avoid confusion we will throughout the paper assume that the optimal solution is unique and that the concept of closest facility (or client) is also uniquely defined. This can be achieved either by slightly perturbing the metric or by simply breaking ties in an arbitrary but fixed way.

When considering a client or facility, it shall be convenient to argue about close clients or facilities. For any $p \in \mathcal{F} \cup \mathcal{C}$ and $r \ge 0$, we therefore define $\mathsf{FBall}_{\mathcal{I}}(p, r) = \{i \in \mathcal{F} : d(p, i) < r\}$ and $\mathsf{CBall}_{\mathcal{I}}(p, r) = \{j \in \mathcal{C} : d(p, j) < r\}$ to be the set of facilities and clients within distance less than $r$ from $p$, respectively. When $\mathcal{I}$ is clear from the context, we omit the subscripts in $\mathsf{cost}_{\mathcal{I}}$, $\mathsf{OPT}_{\mathcal{I}}$, $\mathsf{opt}_{\mathcal{I}}$, $\mathsf{FBall}_{\mathcal{I}}$, and $\mathsf{CBall}_{\mathcal{I}}$.

The standard linear programming relaxation for the $k$-median problem is formulated as follows.

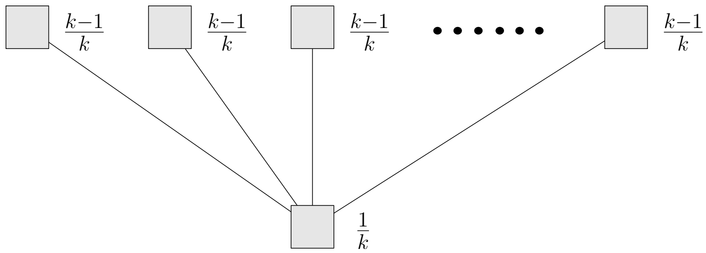


Figure 1: Instance that gives integrality gap $2/(1+1/k)$ and the optimal fractional solution. We have $k+2$ facilities and $k+1$ clients co-located with the top $k+1$ facilities. All edges in the graph have length 1. The optimal integral solution has cost 2, while the optimal fractional solution has cost $(k+1)\left(\frac{k-1}{k}\cdot 0+\frac{1}{k}\cdot 1\right)=1+1/k$.

$$
\begin{aligned}
&\text{minimize } \textstyle\sum_{i\in\mathcal{F},j\in\mathcal{C}} d(i,j)x_{ij} && && \\
&\text{subject to} & \textstyle\sum_{i\in\mathcal{F}} y_i &\le k & & \quad\text{(1a)}\\
& & \textstyle\sum_{i\in\mathcal{F}} x_{ij} &= 1 & j &\in \mathcal{C} \quad\text{(1b)}\\
& & x_{ij} &\le y_i & i\in\mathcal{F}, j &\in \mathcal{C} \quad\text{(1c)}\\
& & x_{ij}, y_i &\in [0,1] & i\in\mathcal{F}, j &\in \mathcal{C} \quad\text{(1d)}
\end{aligned}
$$

Constraint (1a) says that we are allowed to open at most $k$ facilities, Constraint (1b) says that we must connect each client, and Constraint (1c) says that if we connect a client to a facility then that facility has to be opened.

As mentioned earlier, the above linear programming has an integrality gap of 2, even when the underlying metric is a tree. The instance that gives the integrality gap of 2 is depicted in Figure 1. It is a star with $k+1$ leaves. The center of the star is a facility and the leaves are both facilities and clients. Note that a pseudo-solution that opens all leaves, i.e., $k+1$ facilities, has cost 0 whereas any solution that opens only $k$ facilities has cost 2. The solution to the linear program obtained by a linear combination of the pseudo-solution that opens all leaves and the solution that only opens the center of the star has cost $1+1/k$ yielding the integrality gap of 2 when $k$ tends to infinity. In general, a solution that is a linear combination of two pseudo-solutions is called a *bi-point (fractional) solution*. As this concept is important for our pseudo-approximation algorithm, we state its formal definition.

**Definition 2** (bi-point (fractional) solution)**.** *Let $\mathcal{I}=(k,\mathcal{F},\mathcal{C},d)$ be a $k$-median instance. Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two pseudo-solutions to $\mathcal{I}$ such that $|\mathcal{S}_1|\le k<|\mathcal{S}_2|$. Let $a\ge 0, b\ge 0$ be the real numbers such that $a+b=1$ and $a|\mathcal{S}_1|+b|\mathcal{S}_2|=k$. Then, the following fractional solution to $\mathcal{I}$, denoted by $a\mathcal{S}_1+b\mathcal{S}_2$, is called a bi-point (fractional) solution:*

1. $y_i = a1_{i\in\mathcal{S}_1}+b1_{i\in\mathcal{S}_2}$;
2. $x_{i,j}=a1_{\mathsf{clst}(i,\mathcal{S}_1,j)}+b1_{\mathsf{clst}(i,\mathcal{S}_2,j)}$, *where* $\mathsf{clst}(i,\mathcal{S},j)$ *denotes the event that $i$ is the closest facility in $\mathcal{S}$ to $j$.*

It is easy to see that the cost of the fractional solution $a\mathcal{S}_1+b\mathcal{S}_2$ is exactly $a\mathsf{cost}_{\mathcal{I}}(\mathcal{S}_1)+b\mathsf{cost}_{\mathcal{I}}(\mathcal{S}_2)$. Jain and Vazirani [12] gave a *Lagrangian multiplier preserving* 3-approximation for UFL, which immediately yields an algorithm which produces a bi-point solution whose cost is at most 3 times the optimum. Together with an algorithm which converts a bi-point solution to an integral solution at the cost of a factor 2, [12] gave a 6-approximation for $k$-median. Later, the factor 3 was improved by Jain et al. [11] to 2. We now formally state the result of [11].

**Theorem 3** ([11]). *Given a $k$-median instance $\mathcal{I}$, we can find in polynomial time a bi-point solution $a\mathcal{S}_1 + b\mathcal{S}_2$ to $\mathcal{I}$ whose cost is at most 2 times the cost of an optimal solution to $\mathcal{I}$.*

### 1.3 Overview of the Algorithm

The two components of our algorithm are formally stated in Theorem 4 and Theorem 5, whose proofs will be given in Sections 2 and 3, respectively. Together they immediately imply Theorem 1.

**Theorem 4.** *Let $\mathcal{A}$ be a $c$-additive $\alpha$-approximation algorithm for $k$-median, for some $\alpha > 1$. Then, for every $\epsilon > 0$ there is a $\alpha + \epsilon$-approximation algorithm $\mathcal{A}'$ for $k$-median whose running time is $O\left(n^{O(c/\epsilon)}\right)$ times the running time of $\mathcal{A}$.*

**Theorem 5.** *There exists a polynomial time algorithm which, given a $k$-median instance $\mathcal{I} = (k, \mathcal{F}, \mathcal{C}, d)$ and $\epsilon > 0$, produces an $O(1/\epsilon)$-additive $1 + \sqrt{3} + \epsilon$-approximate solution to $\mathcal{I}$.*

We now provide more details about the proof of the two theorems. At first glance, it seems that the transformation from a pseudo-approximation to a real approximation stated in Theorem 4 is impossible, since there are cases where allowing $k + 1$ open facilities would give much smaller cost than only allowing $k$ open facilities. However, we show that we can pre-process the input instance so as to avoid these problematic instances. Roughly speaking, we say that a facility $i$ is dense if the clients in a small ball around $i$ contribute a lot to the cost of the optimum solution $\mathsf{OPT}$ (see Definition 6). We guess the $O(1/\epsilon)$ densest facilities and their respective nearest open facilities in $\mathsf{OPT}$. Then for each such dense facility $i$ whose nearest open facility in $\mathsf{OPT}$ is $i'$, we remove all facilities that are closer to $i$ than $i'$ (including the dense facility $i$). Then we get a residual instance in which the gap between the costs of opening $k + O(1)$ and $k$ facilities is small. The pseudo-approximation algorithm is then applied to this residual instance.

For example, consider the integrality gap instance depicted in Figure 1 and let $\mathsf{OPT}$ be the optimal solution that opens the center and $k - 1$ leaves. Then the two leaves that were not opened contribute a large fraction of the total cost (each contributes $\mathsf{opt}/2$ to be precise) and the two corresponding facilities are dense. By removing these dense facilities in a preprocessing step, the gap between the costs of opening $k + O(1)$ facilities and $k$ facilities for the residual instance becomes small (actually 0 in this example).

Regarding the proof of Theorem 5, we first use Theorem 3 to obtain a bi-point solution for $k$-median whose cost is at most twice the optimum cost. Jain and Vazirani [12] showed how to convert a bi-point solution to an integral solution, losing a multiplicative factor of 2 in the approximation. As we previously mentioned, this factor of 2 is tight, as the fractional solution for the gap instance in Figure 1 is a bi-point solution. Thus, this approach can only yield a 4-approximation.

This is where the $c$-additive pseudo-approximation is used and again the integrality gap instance depicted in Figure 1 inspired our approach. Recall that if we open the $k + 1$ leaves of that instance, then we get a solution of cost 0. In other words, by opening 1 additional facility, we can do better

than the fractional solution. One may argue that this trick is too weak to handle more sophisticated cases and try to enhance the gap instance. A natural way to enhance it is to make many separate copies of the instance to obtain several "stars". One might expect that the fractional cost in each copy is 1, the integral cost in each copy is 2 and opening 1 more facility can only improve the integral solution of one copy and thus does not improve the overall ratio by too much. However, the integral solution can do much better since one cannot restrict the integral solution to open $k$ facilities in each star. As an example, consider the case where we have 2 copies. The integral solution can open $k-1$ facilities in the first star, and $k+1$ facility in the second star. Then, the cost of this solution is 3, as opposed to 4 achieved by opening $k$ facilities in each star. The gap is already reduced to 1.5, without opening additional facilities. Thus, this simple way to enhance the instance failed.

Our pseudo-approximation algorithm is based on this intuition. From the bi-point solution $a\mathcal{F}_1 + b\mathcal{F}_2$, we obtain copies of "stars" (similar to the integrality gap instance). Then for each star we (basically) open either its center with probability $a$ or all its leaves with probability $b$. Note that since either the center or all leaves of a star is open we have that a client always has a "close" facility opened. With this intuition we prove in Section 3 that the expected cost of the obtained pseudo-solution is at most $\frac{1+\sqrt{3}+\epsilon}{2}$ times the cost of the bi-fractional solution if we open $O(1/\epsilon)$ additional facilities. The $O(1/\epsilon)$ additional facilities (and the case distinction in Section 3) comes from the difficulty of handling stars of different sizes. If all stars are of the same size the pseudo-approximation algorithm becomes easier (run the algorithm in Section 3.2 with one group of stars) and one obtains a $\frac{1+\sqrt{3}}{2}$-approximate solution that opens at most $k+3$ facilities.

## 2 Obtain solutions from additive pseudo-solutions

In this section, we prove Theorem 4. As we mentioned earlier, there are instances where pseudo-solutions opening $k+1$ facilities may have much smaller cost than solutions opening $k$ facilities. A key concept to overcome this issue is the notion of sparse instances:

**Definition 6.** *For $A > 0$, an instance $\mathcal{I} = (k, \mathcal{F}, \mathcal{C}, d)$ is $A$-sparse if for each facility $i \in \mathcal{F}$,*

$$(1-\xi)d(i, \mathsf{OPT}_{\mathcal{I}}) \cdot |\mathsf{CBall}_{\mathcal{I}}(i, \xi d(i, \mathsf{OPT}_{\mathcal{I}}))| \leq A, \tag{2}$$

*where $\xi := 1/3$. We shall also say that a facility $i$ is $A$-dense if it violates* (2).

Recall that $d(i, \mathsf{OPT}_{\mathcal{I}})$ is the distance from $i$ to its nearest facility in $\mathsf{OPT}_{\mathcal{I}}$.

The idea of the above definition is to avoid instances where we can significantly reduce the cost by opening $O(1)$ additional facilities. Consider the gap instance $\mathcal{I}$ in Figure 1 and suppose $\mathsf{OPT}_{\mathcal{I}}$ opens the center and the first $k-1$ leaf-facilities. Then $\mathcal{I}$ is not $A$-sparse for $A < \mathsf{opt}_{\mathcal{I}}/2$ since the last two leaf-facilities are $A$-dense.

The usefulness of the definition is twofold. On the one hand, we show that we can concentrate on very sparse instances without loss of generality. On the other hand, we show that any $c$-additive pseudo-solution to a sparse instance can be turned into a solution that opens $k$ facilities by only increasing the cost slightly. The intuition behind the result that we can only concentrate on sparse instances is the following. Consider an instance $\mathcal{I}$ that is not $\mathsf{opt}_{\mathcal{I}}/t$-sparse for some constant $t$. If we consider a facility $i$ that is $\mathsf{opt}_{\mathcal{I}}/t$-dense then the connection cost of the clients contained in $\mathsf{CBall}(i, \xi d(i, \mathsf{OPT}_{\mathcal{I}}))$ in the optimal solution $\mathsf{OPT}_{\mathcal{I}}$ is at least

$(1-\xi)d(i, \mathsf{OPT}_\mathcal{I})|\mathsf{CBall}(i, \xi d(i, \mathsf{OPT}_\mathcal{I}))| > \mathsf{opt}_\mathcal{I}/t$. So, there can essentially (assuming disjointedness of the balls of clients) only be a constant $t$ number of facilities that violate the sparsity condition. We can guess this set of dense facilities, as well as their nearest facility in $\mathsf{OPT}_\mathcal{I}$ in time $n^{O(t)}$.

This is the intuition of Algorithm 1 (that tries to guess and remove $\mathsf{opt}/t$-dense facilities) and the proof of the following lemma which is given in Section 2.1.

**Lemma 7.** *Given a $k$-median instance $\mathcal{I} = (k, \mathcal{F}, \mathcal{C}, d)$ and a positive integer $t$, Algorithm 1 outputs in time $n^{O(t)}$ many $k$-median instances obtained by removing facilities from $\mathcal{I}$ so that at least one, say $\mathcal{I}' = (k, \mathcal{F}' \subseteq \mathcal{F}, \mathcal{C}, d)$, satisfies*

*(7a): the optimal solution $\mathsf{OPT}_\mathcal{I}$ to $\mathcal{I}$ is also an optimal solution to $\mathcal{I}'$; and*

*(7b): $\mathcal{I}'$ is $\mathsf{opt}_\mathcal{I}/t$-sparse.*

Note that $\mathcal{I}'$ is obtained by removing facilities from $\mathcal{I}$. Therefore any solution to $\mathcal{I}'$ defines a solution to $\mathcal{I}$ of the same cost and we can thus restrict our attention to sparse instances. The next lemma shows the advantage of considering such instances. Assume we now have a $c$-additive solution $\mathcal{T}$ to a sparse instance $\mathcal{I}$. Algorithm 2 tries first in Lines 2-3 to identify facilities in $\mathcal{T}$ whose removal does not increase the cost by too much. If the removal results in a set of at most $k$ facilities, we have obtained a "good" solution returned at Step 4 of the algorithm. Otherwise, as we prove in Section 2.2 using sparsity, more than $k-t$ of the facilities of the solution $\mathcal{T}$ are very close to facilities in $\mathsf{OPT}_\mathcal{I}$. Algorithm 2 therefore tries to guess these facilities (the set $\mathcal{D}$) and the remaining facilities of $\mathsf{OPT}_\mathcal{I}$ (the set $\mathcal{V}$). The obtained bounds are given in the following lemma.

**Lemma 8.** *Given an $A$-sparse instance $\mathcal{I} = (k, \mathcal{F}, \mathcal{C}, d)$, a $c$-additive pseudo-solution $\mathcal{T}$, $\delta \in (0, 1/8)$, and an integer $t \geq 2c/(\delta\xi)$, Algorithm 2 finds in time $n^{O(t)}$ a set $\mathcal{S} \subseteq \mathcal{F}$ such that:*

*(8a): $\mathcal{S}$ is a solution to $\mathcal{I}$, i.e, $|\mathcal{S}| \leq k$; and*

*(8b): $\mathsf{cost}_\mathcal{I}(\mathcal{S}) \leq \max\left\{\mathsf{cost}_\mathcal{I}(\mathcal{T}) + cB, \frac{1+3\delta}{1-3\delta} \cdot \mathsf{opt}_\mathcal{I}\right\}$, where $B := 2 \cdot \frac{A + \mathsf{cost}_\mathcal{I}(\mathcal{T})/t}{\xi\delta}$.*

Before giving the proofs of Lemmas 7 and 8 let us see how they imply the main result of this section.

**Proof of Theorem 4.** Select the largest $\delta \in (0, 1/8)$ such that $(1+3\delta)/(1-3\delta) \leq \alpha$ and $t := \frac{4}{\epsilon} \cdot \frac{\alpha c}{\xi \cdot \delta} = O(c/\epsilon)$. Given a $k$-median instance $\mathcal{I}$, use Algorithm 1 to obtain a set of $k$-median instances such that at least one of these instances, say $\mathcal{I}'$, satisfies the properties of Lemma 7. In particular, $\mathcal{I}'$ is $\mathsf{opt}_\mathcal{I}/t$-sparse. Now use algorithm $\mathcal{A}$ to obtain $c$-additive pseudo-solutions to each of these instances. Note that when we apply $\mathcal{A}$ to $\mathcal{I}'$, we obtain a solution $\mathcal{T}$ such that $\mathsf{cost}_\mathcal{I}(\mathcal{T}) = \mathsf{cost}_{\mathcal{I}'}(\mathcal{T}) \leq \alpha \cdot \mathsf{opt}_{\mathcal{I}'} = \alpha \cdot \mathsf{opt}_\mathcal{I}$. Finally, use Algorithm 2 (with the same $t$ and $\delta$ selected as above) to transform the pseudo-solutions into real solutions and return the solution to $\mathcal{I}$ of minimum cost. The cost of the returned solution is at most the cost of $\mathcal{S}$ where $\mathcal{S}$ is the solution obtained by transforming $\mathcal{T}$. By Lemmas 7 and 8, we have that $\mathsf{cost}_\mathcal{I}(S) = \mathsf{cost}_{\mathcal{I}'}(S)$ is at most

$$\max\left\{\mathsf{cost}_\mathcal{I}(\mathcal{T}) + c \cdot 2\frac{\mathsf{opt}_\mathcal{I} + \mathsf{cost}_\mathcal{I}(\mathcal{T})}{t\xi\delta}, \frac{1+3\delta}{1-3\delta}\mathsf{opt}_\mathcal{I}\right\},$$

which in turn, by the selection of $\delta, \xi$, and $t$, is at most $\alpha\mathsf{opt}_\mathcal{I} + c \cdot \frac{4\alpha\mathsf{opt}_\mathcal{I}}{t\xi\delta} \leq (\alpha+\epsilon)\mathsf{opt}_\mathcal{I}$.

We conclude the proof of Theorem 4 by observing that the runtime of the algorithm is $n^{O(t)} = n^{O(c/\epsilon)}$ times the runtime of $\mathcal{A}$.

**Input:** a $k$-median instance $\mathcal{I} = (k, \mathcal{F}, \mathcal{C}, d)$ and a positive integer $t$
**Output:** a set of $k$-median instances so that at least one satisfies the properties of Lemma 7

**for all** $t' \leq t$ facility-pairs $(i_1, i'_1), (i_2, i'_2), \ldots, (i_{t'}, i'_{t'})$ **output** $(k, \mathcal{F}', \mathcal{C}, d)$, where

$$\mathcal{F}' = \mathcal{F} \setminus \bigcup_{z=1}^{t'} \mathsf{FBall}(i_z, d(i_z, i'_z)) \qquad \triangleright \text{ the facilities that are closer to } i_z \text{ than } i'_z \text{ is to } i_z \text{ are removed}$$

**Algorithm 1:** Enumeration of $k$-median instances.

## 2.1 Proof of Lemma 7: obtaining a sparse instance

First note that Algorithm 1 selects $n^{O(t)}$ facility-pairs and can be implemented to run in time $n^{O(t)}$. We proceed by showing that for one selection of facility-pairs the obtained instance satisfies the properties of Lemma 7. Consider a maximal-length sequence $(i_1, i'_1), (i_2, i'_2), \ldots, (i_\ell, i'_\ell)$ of facility-pairs satisfying: for every $b = 1, \ldots, \ell$,

- $i_b \in \mathcal{F} \setminus \bigcup_{z=1}^{b-1} \mathsf{FBall}(i_z, d(i_z, i'_z))$ is an $\mathsf{opt}_\mathcal{I}/t$-dense facility; and
- $i'_b$ is the closest facility to $i_b$ in $\mathsf{OPT}_\mathcal{I}$.

Note that the instance $\mathcal{I}' := (k, \mathcal{F}', \mathcal{C}, d)$ with $\mathcal{F}' = \mathcal{F} \setminus \bigcup_{z=1}^{\ell} \mathsf{FBall}(i_z, d(i_z, i'_z))$ is $\mathsf{opt}_\mathcal{I}/t$-sparse since otherwise the sequence $(i_1, i'_1), (i_2, i'_2), \ldots, (i_\ell, i'_\ell)$ would not be of maximal length. Moreover, since we do not remove any facilities in $\mathsf{OPT}_\mathcal{I}$, i.e., $(\mathcal{F} \setminus \mathcal{F}') \cap \mathsf{OPT}_\mathcal{I} = \emptyset$, we have that $\mathsf{OPT}_\mathcal{I}$ is also an optimal solution to $\mathcal{I}'$. In other words, $\mathcal{I}'$ satisfies the properties of Lemma 7.

We complete the proof by showing that Algorithm 1 enumerates $\mathcal{I}'$, i.e., that $\ell \leq t$. For the sake of notation let $\mathcal{B}_z := \mathsf{CBall}(i, \xi d(i_z, i'_z))$. First, note that the client-balls $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_\ell$ are disjoint. Indeed, if a ball $\mathcal{B}_z$ overlaps a ball $\mathcal{B}_w$ with $1 \leq z < w \leq \ell$ then $d(i_z, i_w) < \xi d(i_z, i'_z) + \xi d(i_w, i'_w)$. However, since $i_w$ must be in $\mathcal{F} \setminus \mathsf{FBall}(i_z, d(i_z, i'_z))$, we have $d(i_z, i_w) \geq d(i_z, i'_z)$. Since $i'_w$ is the closest facility in $\mathsf{OPT}_\mathcal{I}$ to $i_w$, we have $d(i_w, i'_w) \leq d(i_w, i'_z)$, which, by triangle inequalities, is at most $d(i_z, i_w) + d(i_z, i'_z) \leq 2d(i_z, i_w)$. Hence (using that $\xi = 1/3$),

$$\xi(d(i_z, i'_z) + d(i_w, i'_w)) \leq 3\xi d(i_z, i_w) \leq d(i_z, i_w),$$

which implies that the balls do not overlap.

Second, note that the connection cost of a client in $\mathcal{B}_z$ is, by triangle inequalities, at least $(1 - \xi)d(i_z, i'_z) = (1 - \xi)d(i_z, \mathsf{OPT}_\mathcal{I})$. We thus have (using that the client-balls are disjoint) that $\mathsf{opt}_\mathcal{I} \geq \sum_{z=1}^{\ell}(1 - \xi)d(i_z, \mathsf{OPT}_\mathcal{I})|\mathcal{B}_z|$. As we only selected $\mathsf{opt}_\mathcal{I}/t$-dense facilities, $(1 - \xi)d(i_z, \mathsf{OPT}_\mathcal{I})|\mathcal{B}_z| \geq \mathsf{opt}_\mathcal{I}/t$ and hence $\mathsf{opt}_\mathcal{I} \geq \ell \mathsf{opt}_\mathcal{I}/t$. It follows that $t \geq \ell$ which completes the proof of Lemma 7.

## 2.2 Proof of Lemma 8: obtain solution to sparse instance from pseudo-solution

We start by analyzing the running time of Algorithm 2. Clearly the while loop can run at most $c$ iterations (a constant). The number of different pairs $(\mathcal{D}, \mathcal{V})$ in the for loop is at most

$$\sum_{\ell=0}^{t} \binom{|\mathcal{T}'|}{k - \ell} \binom{|\mathcal{F}|}{\ell}.$$

Notice that $|\mathcal{T}'| \leq k + c$ and $c \leq t$. For sufficiently large $k$ and $|\mathcal{F}|$, the above quantity is at most $\binom{|\mathcal{F}|}{t} \sum_{\ell=0}^{t} \binom{k+c}{c+\ell} = n^{O(t)}$. Algorithm 2 can thus be implemented to run in time $n^{O(t)}$ as required. Moreover, it is clear from its definition that it always returns a solution $\mathcal{S}$, i.e., $|\mathcal{S}| \leq k$.

**Input:** an $A$-sparse instance $\mathcal{I} = (k, \mathcal{F}, \mathcal{C}, d)$, a $c$-additive pseudo-solution $\mathcal{T}$, an integer $t \geq c$ and $\delta \in (0, 1/8)$

**Output:** A solution $\mathcal{S}$ satisfying the properties of Lemma 8

1: $\mathcal{T}' := \mathcal{T}$ and $B := 2 \cdot \frac{A + \mathsf{cost}_{\mathcal{I}}(\mathcal{T})/t}{\delta\xi}$

2: **while** $|\mathcal{T}'| > k$ and there is a facility $i \in \mathcal{T}'$ such that $\mathsf{cost}_{\mathcal{I}}(\mathcal{T}' \setminus \{i\}) \leq \mathsf{cost}_{\mathcal{I}}(\mathcal{T}') + B$ **do**

3: Remove $i$ from $\mathcal{T}'$;

4: **return** $\mathcal{S} := \mathcal{T}'$ if $|\mathcal{T}'| \leq k$;

5: **for all** $\mathcal{D} \subseteq \mathcal{T}'$ and $\mathcal{V} \subseteq \mathcal{F}$ such that $|\mathcal{D}| + |\mathcal{V}| = k$ and $|\mathcal{V}| < t$ **do**

6: For $i \in \mathcal{D}$, let $L_i = d(i, \mathcal{T}' \setminus \{i\})$ and $f_i$ be the facility in $\mathsf{FBall}(i, \delta L_i)$ that minimizes

$$\sum_{j \in \mathsf{CBall}(i, L_i/3)} \min\{d(f_i, j), d(j, \mathcal{V})\}$$

7: Let $\mathcal{S}_{\mathcal{D},\mathcal{V}} := \mathcal{V} \cup \{f_i : i \in \mathcal{D}\}$

8: **return** $\mathcal{S} := \arg\min_{S_{\mathcal{D},\mathcal{V}}} \mathsf{cost}_{\mathcal{I}}(S_{\mathcal{D},\mathcal{V}})$

**Algorithm 2:** Obtaining a solution from a $c$-additive pseudo-solution.

We proceed by proving that $\mathcal{S}$ satisfies (8b) of Lemma 8. Suppose first that the algorithm returns at Line 4. By the condition of the while loop from Line 2 to 3, we increase $\mathsf{cost}_{\mathcal{I}}(\mathcal{T}')$ by at most $B$ each time we remove an element from $\mathcal{T}'$. We remove at most $c$ elements and thus we increase the total cost by at most $cB$. It follows that (8b) is immediately satisfied in this case.

From now on suppose instead that we reached Line 5 of Algorithm 2 and thus $|\mathcal{T}'| > k$. We shall exhibit sets $\mathcal{D}_0$ and $\mathcal{V}_0$ such that $|\mathcal{D}_0| + |\mathcal{V}_0| = k$, $|\mathcal{V}_0| < t$ and $\mathsf{cost}(\mathcal{S}_{\mathcal{D}_0,\mathcal{V}_0}) \leq \frac{1+3\delta}{1-3\delta}\mathsf{opt}_{\mathcal{I}}$. As Algorithm 2 selects $\mathcal{D}_0$ and $\mathcal{V}_0$ in one iteration and it returns the minimum cost solution, this concludes the proof of Lemma 8. In order to define the sets $\mathcal{D}_0$ and $\mathcal{V}_0$ it shall be convenient to use the following definitions.

**Definition 9.** *For every facility $i \in \mathcal{T}'$, let $L_i = d(i, \mathcal{T}' \setminus \{i\})$ be the distance from $i$ to its nearest neighbor in $\mathcal{T}'$, and let $\ell_i = d(i, \mathsf{OPT}_{\mathcal{I}})$ be the minimum distance from $i$ to any facility in $\mathsf{OPT}_{\mathcal{I}}$.*

*For a facility $i \in \mathcal{T}'$, we say $i$ is* determined *if $\ell_i < \delta L_i$. Otherwise, we say $i$ is* undetermined.

The sets $\mathcal{D}_0$ and $\mathcal{V}_0$ are now defined as follows. Set $\mathcal{D}_0$ contain all facilities in $i \in \mathcal{T}'$ that are determined. If we let $f_i^*$ for $i \in \mathcal{D}_0$ be the facility in $\mathsf{OPT}_{\mathcal{I}}$ that is closest to $i$, then set $\mathcal{V}_0 := \mathsf{OPT}_{\mathcal{I}} \setminus \{f_i^* : i \in \mathcal{D}_0\}$. The intuition of $\mathcal{D}_0$ and $\mathcal{V}_0$ is that the solution $\mathcal{S}_{\mathcal{D}_0,\mathcal{V}_0}$ is very close to $\mathsf{OPT}_{\mathcal{I}}$: the only difference is the selection of $f_i$ at Line 6 of Algorithm 2 instead of $f_i^*$. Since each $i \in \mathcal{D}_0$ is determined, selecting $f_i$ greedily using a "locally" optimal strategy gives a good solution.

We first show that sets $\mathcal{D}_0$ and $\mathcal{V}_0$ are indeed selected by Algorithm 2 and then we conclude the proof of the lemma by bounding the cost of $\mathcal{S}_{\mathcal{D}_0,\mathcal{V}_0}$.

**Claim 10.** *$|\mathcal{D}_0| + |\mathcal{V}_0| = k$ and $|\mathcal{V}_0| < t$.*

*Proof of Claim.* We start by proving that $|\mathcal{D}_0| + |\mathcal{V}_0| = k$. Recall that $\mathcal{V}_0 = \mathsf{OPT}_{\mathcal{I}} \setminus \{f_i^* : i \in \mathcal{D}_0\}$. It is not hard to see that $f_i^* \neq f_{i'}^*$ for two distinct facilities in $\mathcal{D}_0$. This is indeed true since $d(i, i') \geq \max(L_i, L_{i'}), d(i, f_i^*) \leq \delta L_i, d(i', f_{i'}^*) \leq \delta L_{i'}$ and $\delta \leq 1/8$. Thus, $f^*(\mathcal{D}_0) := \{f_i^* : i \in \mathcal{D}_0\}$ has size

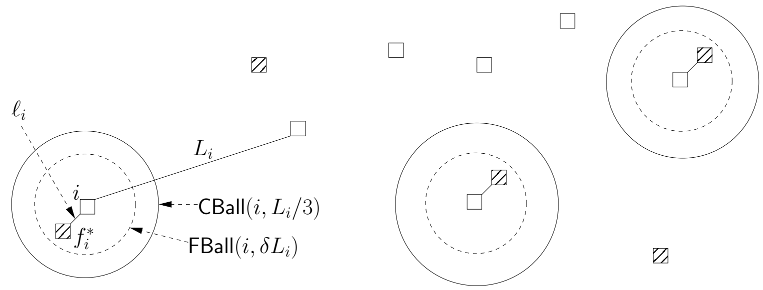


Figure 2: Definitions of $\mathcal{D}_0, \mathcal{V}_0$ and $\mathcal{U}_0$. Dashed and empty squares represent facilities in $\mathsf{OPT}_{\mathcal{I}}$ and $\mathcal{T}'$ respectively. $\mathcal{D}_0$ is the set of empty squares circles. A dashed circle represents $\mathsf{FBall}(i, \delta L_i)$ for a determined facility $i \in \mathcal{D}_0$. Thus, $f_i^*$ is in the ball since $\ell_i < \delta L_i$. $\mathcal{U}_0(\mathcal{V}_0$, resp.) is the sets of empty (dashed, resp.) squares that are not inside any circle. A solid circle for $i \in \mathcal{D}_0$ represents the "care-set" of $i$.

$|\mathcal{D}_0|$, which in turn implies that (to simplify calculations we assume w.l.o.g. that $|\mathsf{OPT}_{\mathcal{I}}| = k$)

$$|\mathcal{V}_0| = |\mathsf{OPT}_{\mathcal{I}}| - |\mathcal{D}_0| = k - |\mathcal{D}_0|.$$

We proceed by proving $|\mathcal{V}_0| < t$. Note that the sets of determined and undetermined facilities partition $\mathcal{T}'$. Therefore, if we let $\mathcal{U}_0$ be the set of undetermined facilities, we have that $|\mathcal{D}_0| = |\mathcal{T}'| - |\mathcal{U}_0|$. Combining this with the above expression for $|\mathcal{V}_0|$ gives us

$$|\mathcal{V}_0| = k - |\mathcal{T}'| + |\mathcal{U}_0| \le |\mathcal{U}_0|.$$

We complete the proof of the claim by showing that $|\mathcal{U}_0| < t$.

By the assumption that we reached Line 5 of Algorithm 2, we have $|\mathcal{T}'| > k$ and $\mathsf{cost}_{\mathcal{I}}(\mathcal{T}' \setminus \{i\}) > \mathsf{cost}_{\mathcal{I}}(\mathcal{T}') + B$ for every $i \in \mathcal{T}'$. Assume towards contradiction that $|\mathcal{U}_0| \ge t$. For every $i \in \mathcal{T}'$, let $\mathcal{C}_i$ be the set of clients in $\mathcal{C}$ connected to $i$ in the solution $\mathcal{T}'$ and $C_i$ be the total connection cost of these clients. Thus, $\mathsf{cost}_{\mathcal{I}}(\mathcal{T}') = \sum_{i \in \mathcal{T}'} C_i$. Take the facility $i \in \mathcal{U}_0$ with the minimum $C_i$. Then, we have $C_i \le \mathsf{cost}_{\mathcal{I}}(\mathcal{T}')/t$. Let $i'$ be the nearest neighbor of $i$ in $\mathcal{T}'$; thus $d(i, i') = L_i$.

We shall remove the facility $i$ from $\mathcal{T}'$ and connect the clients in $\mathcal{C}_i$ to $i'$. In order to consider incremental connection cost incurred by the operation, we divide $\mathcal{C}_i$ into two parts.

**$\mathcal{C}_i \cap \mathsf{CBall}(i, \delta\xi L_i)$.** Since $i$ is undetermined, we have $\delta L_i \le \ell_i$ and $\mathsf{CBall}(i, \delta\xi L_i) \subseteq \mathsf{CBall}(i, \xi \ell_i)$. As $\mathcal{I}$ is an $A$-sparse instance, $i$ is not an $A$-dense facility. That is $(1 - \xi)\,|\mathsf{CBall}(i, \xi \ell_i)|\,\ell_i \le A$, implying

$$(1 + \delta\xi)\,|\mathcal{C}_i \cap \mathsf{CBall}(i, \delta\xi L_i)|\,L_i \le \frac{(1 + \delta\xi)}{\delta(1 - \xi)} A \le A/(\delta\xi).$$

Then, as each client in $\mathcal{C}_i \cap \mathsf{CBall}(i, \delta\xi L_i)$ has distance at most $(1 + \delta\xi)L_i$ to $i'$ (by triangle inequalities), connecting all clients in $\mathcal{C}_i \cap \mathsf{CBall}(i, \delta\xi L_i)$ to $i'$ can cost at most $A/(\delta\xi)$.

**$\mathcal{C}_i \setminus \mathsf{CBall}(i, \delta\xi L_i)$.** Consider any client $j$ in this set. Since $d(j, i') \le d(j, i) + L_i$ and $d(j, i) \ge \delta\xi L_i$, we have $\frac{d(j,i') - d(j,i)}{d(j,i)} \le \frac{L_i}{\delta\xi L_i} = 1/(\delta\xi)$. Hence, the connection cost of a single client is increased by at most a factor $1/(\delta\xi)$. Therefore, the total connection cost increases by at most $C_i/(\delta\xi)$, which by the selection of $i$ is at most $\mathsf{cost}_{\mathcal{I}}(\mathcal{T}')/(\delta\xi t)$.

Summing up the two quantities, removing $i$ from $\mathcal{T}'$ can only increase the connection cost by at most $\frac{A+\mathsf{cost}_{\mathcal{I}}(\mathcal{T}')/t}{\delta\xi}$. As the while loop of Algorithm 2 ran for less than $c$ iterations, $\mathsf{cost}_{\mathcal{I}}(\mathcal{T}') < \mathsf{cost}_{\mathcal{I}}(\mathcal{T}) + cB$. Therefore, $\frac{A+\mathsf{cost}_{\mathcal{I}}(\mathcal{T}')/t}{\delta\xi} < \frac{A+(\mathsf{cost}_{\mathcal{I}}(\mathcal{T})+cB)/t}{\delta\xi}$ which since $t \geq 2c/(\delta\xi)$ is at most $\frac{A+\mathsf{cost}_{\mathcal{I}}(\mathcal{T})/t}{\delta\xi} + B/2 = B$ leading to a contradiction. Hence, $|\mathcal{U}_0| < t$ which concludes the proof of the claim. □

Having proved that the instance $\mathcal{S}_{\mathcal{D}_0,\mathcal{V}_0}$ is selected by Algorithm 2, we conclude the proof of Lemma 8 by bounding the cost of $\mathcal{S}_{\mathcal{D}_0,\mathcal{V}_0}$.

Recall that, for every $i \in \mathcal{D}_0$, Algorithm 2 opens one facility $f_i$ in the ball $\mathsf{FBall}(i, \delta L_i)$. We know we can do this so that the connection cost of $\mathcal{C}$ is $\mathsf{opt}_{\mathcal{I}}$. We show that we can approximate this instance within a factor of $1+O(\delta)$. Roughly speaking, if a client is far away from any of these balls, then it does not care which facilities to open inside the balls, up to a factor $1+O(\delta)$. If a client is close to one of these balls, say $\mathsf{FBall}(i, \delta L_i)$, then we put the client into the "care-set" of $i$. For each $i$, we open a facility in the ball that is best for its care-set.

To be more specific, let the care-set of $i$ be $\mathsf{CBall}(i, L_i/3)$ for any $i \in \mathcal{D}_0$. Clearly, the balls $\mathsf{CBall}(i, L_i/3), i \in \mathcal{D}_0$ are disjoint. As stated in Line 6 of Algorithm 2, we open a facility $f_i$ in $\mathsf{FBall}(i, \delta L_i)$ that minimizes

$$\sum_{j \in \mathsf{CBall}(i, L_i/3)} \min\left\{d(f_i, j), d(j, \mathcal{V}_0)\right\}.$$

**Claim 11.** $\mathsf{cost}_{\mathcal{I}}(\mathcal{S}_{\mathcal{D}_0,\mathcal{V}_0}) \leq \frac{1+3\delta}{1-3\delta}\mathsf{opt}_{\mathcal{I}}$.

*Proof of Claim.* We compare $\mathsf{OPT}_{\mathcal{I}}$ and $\mathcal{S}_{\mathcal{D}_0,\mathcal{V}_0}$. Consider a client $j \in \mathsf{CBall}(i, L_i/3)$ for some $i \in \mathcal{D}_0$. The distance from $j$ to any facility in $\mathsf{FBall}(i, \delta L_i)$ is at most $(1/3+\delta)L_i$. For any distinct facility $i' \in \mathcal{D}_0$, the distance from $j$ to any facility in $\mathsf{FBall}(i', \delta L_{i'})$ is at least $d(i,i') - L_i/3 - \delta L_{i'} \geq d(i,i') - d(i,i')/3 - \delta d(i,i') = (2/3-\delta)d(i,i') \geq (2/3-\delta)L_i$. For $\delta \leq 1/8$, $1/3+\delta < 2/3-\delta$. Thus, $j$ is either connected to $f_i^*$ or some facility in $\mathcal{V}_0$ in the solution $\mathsf{OPT}_{\mathcal{I}}$. Noticing that we are selecting the best $f_i$ for every $i \in \mathcal{D}_0$, the total connection cost of $\bigcup_{i\in\mathcal{D}_0} \mathsf{CBall}(i, L_i/3)$ in the solution $\mathcal{S}_{\mathcal{D}_0,\mathcal{V}_0}$ is at most that in $\mathsf{OPT}_{\mathcal{I}}$.

Now, consider a client $j$ that is not in $\bigcup_{i\in\mathcal{D}_0} \mathsf{CBall}(i, L_i/3)$. If it is connected to some facility in $\mathcal{V}_0$ in the solution $\mathsf{OPT}_{\mathcal{I}}$, then the connection cost of $j$ in the solution $\mathcal{S}_{\mathcal{D}_0,\mathcal{V}_0}$ can not be larger, since $\mathcal{V}_0 \subseteq \mathcal{S}$. Assume $j$ is connected to $f_i^* \in \mathsf{CBall}(i, L_i/3)$ for some $i \in \mathcal{D}_0$. We compare $d(j, f_i^*)$ to $d(j, f_i)$:

$$\frac{d(j,f_i)}{d(j,f_i^*)} \leq \frac{d(j,i)+\delta L_i}{d(j,i)-\delta L_i} \leq \frac{L_i/3+\delta L_i}{L_i/3-\delta L_i} = \frac{1+3\delta}{1-3\delta}.$$

Thus, $\mathcal{S}_{\mathcal{D}_0,\mathcal{V}_0}$ has connection cost at most $\frac{1+3\delta}{1-3\delta}\mathsf{opt}_{\mathcal{I}}$. □

## 3 An $O(1/\epsilon)$-additive $1+\sqrt{3}+\epsilon$ approximation for $k$-median

This section is dedicated to prove Theorem 5. Given a $k$-median instance $\mathcal{I} = (k, \mathcal{F}, \mathcal{C}, d)$, we first use Theorem 3 to obtain a bi-point solution $a\mathcal{S}_1 + b\mathcal{S}_2$ whose cost is at most 2 times the optimum cost of $\mathcal{I}$. Then it suffices to convert $a\mathcal{S}_1 + b\mathcal{S}_2$ into an $O(1/\epsilon)$-additive solution, whose cost is at most $\frac{1+\sqrt{3}+\epsilon}{2}$ times that of $a\mathcal{S}_1 + b\mathcal{S}_2$.

By the definition of bi-point solutions, we have $a+b=1$, $|\mathcal{F}_1| \le k < |\mathcal{F}_2|$ and $a|\mathcal{F}_1|+b|\mathcal{F}_2| = k$. It shall be convenient to think of $a\mathcal{F}_1 + b\mathcal{F}_2$ as a bipartite graph (see Figure 3) with vertex sets $\mathcal{F}_1$ and $\mathcal{F}_2$ and an edge for each client $j \in \mathcal{C}$ that is incident to its closest facilities in $\mathcal{F}_1$ and $\mathcal{F}_2$ denoted by $i_1(j)$ and $i_2(j)$, respectively. Moreover, let $d_1(j) := d(j, i_1(j))$ and $d_2(j) := d(j, i_2(j))$. Then, the (fractional) connection cost of $j$ in the bi-point solution is $ad_1(j) + bd_2(j)$. Similarly, if we let $d_1 := \mathsf{cost}(\mathcal{F}_1) = \sum_{j\in\mathcal{C}} d_1(j)$ and $d_2 := \mathsf{cost}(\mathcal{F}_2) = \sum_{j\in\mathcal{C}} d_2(j)$ then the bi-point solution has cost $ad_1 + bd_2$.

We shall prove Theorem 5 by exhibiting different algorithms based on the value of $a$. Specifically, we shall distinguish between the cases when $a$ is in $\left(0, \frac{\sqrt{3}-1}{4}\right]$, $\left(\frac{\sqrt{3}-1}{4}, \frac{2}{1+\sqrt{3}}\right]$, and $\left(\frac{2}{1+\sqrt{3}}, 1\right]$. The simplest case is when $a \in \left(\frac{2}{1+\sqrt{3}}, 1\right]$: the solution where we open all facilities in $\mathcal{F}_1$ is then a $\frac{d_1}{ad_1+bd_2} \le 1/a = (1+\sqrt{3})/2$ approximation.

For the two remaining cases, we will use the concept of *stars*. For each facility $i \in \mathcal{F}_2$ define $\pi(i)$ to be the facility in $\mathcal{F}_1$ that is closest to $i$. For a facility $i \in \mathcal{F}_1$ , let $S_i = \{i' \in \mathcal{F}_2 : \pi(i') = i\}$. We think of $S_i$ as a star with *center* $i$ and *leaves* $S_i$. Note that by the definition of stars, we have that any client $j$ with $i_2(j) \in S_i$ has $d(i_2(j), i) \le d(i_2(j), i_1(j)) = d_2(j) + d_1(j)$ and therefore $d(j,i) \le d(j, i_2(j)) + d(i_2(j), i) \le 2d_2 + d_1$. Our algorithms will ensure that there is an open facility "close" to every client by always opening $i$ if not all facilities in $S_i$ are opened. The strategy for either opening the center of a star or its leaves (or sometimes both) depends on the value of $a$. We start in Section 3.1 by explaining the simpler case when $a \in \left(0, \frac{\sqrt{3}-1}{4}\right]$ and then complete the proof of Theorem 5 by considering the final case in Section 3.2.

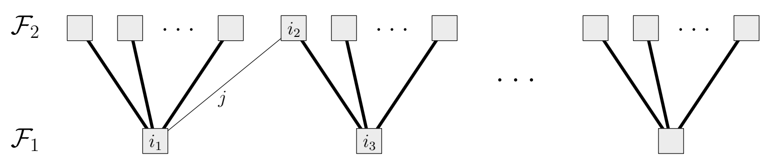


Figure 3: Depiction of the bipartite graph associated to a bi-point solution. The fat edges are the clients that form the edges of the stars. For clarity, we only depicted one client $j$ that is not part of a star. Client $j$ has distances $d(j, i_1) = d_1(j), d(j, i_2) = d_2$ and $d(j, i_3) \le 2d_2(j) + d_1(j)$.

### 3.1 Algorithm for $a \in \left(0, \frac{\sqrt{3}-1}{4}\right]$

The idea behind our algorithm is that when $a$ is small then we can open most facilities in $\mathcal{F}_2$. We shall do so by starting with the trivial solution $\mathcal{F}_1$ that we will improve by almost greedily selecting stars and open all their leaves while closing their centers.

As we will maintain the property that $i$ is open if not all facilities in $S_i$ are open, we have that the connection cost of a client $j$ is $d_2(j)$ if $i_2(j)$ is open and at most $d_1(j) + 2d_2(j)$ otherwise. Consider the trivial solution where we open all facilities in $\mathcal{F}_1$. Then the total connection cost is upper-bounded by $\sum_j (d_1(j) + 2d_2(j))$. If we open the facilities in $S_i$ instead of $i$ this will save us the cost $\sum_{j\in\delta(S_i)}(d_1(j) + d_2(j))$, where $\delta(S_i)$ denotes the clients that are incident to the facilities in $S_i$. This motivates the following linear program that maximizes the cost we will save compared

to the trivial solution:

$$\max \sum_{i \in \mathcal{F}_1} \sum_{j \in \delta(S_i)} (d_1(j) + d_2(j)) x_i \qquad \text{subject to}$$

$$\sum_{i \in \mathcal{F}_1} x_i(|S_i| - 1) \leq k - |\mathcal{F}_1|$$
$$0 \leq x_i \leq 1, \quad \forall i \in \mathcal{F}_1$$

Intuitively, $x_i$ takes value 1 if we open all the facilities in $S_i$ and 0 if we open $i$. If $x_i = 1$, we need to open $|S_i| - 1$ more facilities (i.e, close $i$ and open all facilities in $S_i$). Thus, the constraint says that we can only open $k$ facilities. Note that this is a Knapsack LP and hence it is easy to see that an optimal solution has at most one fractional variable. Furthermore, $x_i = b$ is a feasible solution since $b|\mathcal{F}_2| - b|\mathcal{F}_1| = k - |\mathcal{F}_1|$. Therefore, the optimal solution to the LP has value at least $b(d_1 + d_2)$.

Consider an optimal solution to the Knapsack LP with at most 1 fractional variable. Then, we open all the facilities in $S_i$ with $x_i = 1$, all the facilities $i \in \mathcal{F}_1$ with $x_i = 0$, and for the $i$ with fractional $x_i$ we open $i$ and $\lceil x_i |S_i| \rceil$ facilities in $S_i$ uniformly at random. (This step can easily be derandomized by greedily selecting the $\lceil x_i |S_i| \rceil$ facilities in $S_i$ that maximizes the reduced cost.)

Note that we opened the facilities so that the (expected) saved cost compared to the trivial solution is at least the value of the optimal solution to the linear program. Therefore, this gives us a solution of (expected) cost at most $2d_2 + d_1 - b(d_2 + d_1) = (1 + a)d_2 + ad_1$. Also, the solution opens at most $k + 2$ facilities, where the additive term 2 comes from the star $S_i$ with fractional $x_i$ value.

Since we can assume that $d_2 \leq d_1$ (otherwise we can simply open all facilities in $\mathcal{F}_1$), the algorithm has an approximation guarantee of

$$\frac{(1 + a)d_2 + ad_1}{(1 - a)d_2 + ad_1} \leq (1 + 2a),$$

which is at most $\frac{1+\sqrt{3}}{2}$ if $a \leq \frac{\sqrt{3}-1}{4}$.

## 3.2 Algorithm for $a \in \left(\frac{\sqrt{3}-1}{4}, \frac{2}{1+\sqrt{3}}\right]$

In this subsection, we give the algorithm for the most complex case. To simplify the arguments, we give a randomized algorithm that can easily be derandomized using the standard method of conditional probabilities. The idea is that we wish to describe a randomized rounding that opens a facility in $\mathcal{F}_1$ with probability $\approx a$ and a facility in $\mathcal{F}_2$ with probability $\approx b$ and at the same time ensuring that there always is an open facility “close” to a client by maintaining the property: if $i$ is not open then all facilities in $S_i$ are open for all stars.

We now describe such a randomized rounding that takes a parameter $\eta > 0$ that balances the achieved approximation guarantee with the amount of additional facilities we open: the achieved approximation ratio is $(1+\eta)\frac{1+\sqrt{3}}{2}$ while we open at most $k+O(1/\eta)$ facilities. It shall be convenient to distinguish between large and small stars. We say that a star $S_i$ is *large* if $|S_i| \geq 2/(ab\eta)$ and

*small* otherwise. Moreover, we partition the small stars into $\lceil 2/(ab\eta) \rceil$ groups according to their sizes:

$$\mathcal{U}_h = \{i \in \mathcal{F}_1 : |S_i| = h\} \qquad \text{for } h = 0, 1, \ldots, \lceil 2/(ab\eta) \rceil - 1.$$

The randomized algorithm can now be described as follows:

1: For each large star $S_i$: open $i$ and open $\lfloor b(|S_i| - 1) \rfloor$ facilities in $S_i$ uniformly at random.

2: For each group $\mathcal{U}_h$ of small stars: take a random permutation of the stars in $\mathcal{U}_h$, open the centers of the first $\lceil a|\mathcal{U}_h| \rceil + 1$ stars, and open all leaves of the remaining stars. In addition, if we let $L$ be the number of already opened leaves subtracted from $bh|\mathcal{U}_h|$, then with probability $\lceil L \rceil - L$ open $\lfloor L \rfloor$ and with remaining probability open $\lceil L \rceil$ randomly picked leaves in the first $\lceil a|\mathcal{U}_h| \rceil + 1$ stars.

Note that for a large star the algorithm always opens its center and (almost) a $b$ fraction of its leaves. For a group $\mathcal{U}_h$ of small stars, note that we open either the center (with probability at least $a$) or all leaves of a star. Moreover, we open the additional leaves so that in expectation exactly a $b$ fraction of the leaves of the stars in $\mathcal{U}_h$ are opened.

We start by showing that the algorithm does not open too many facilities; we then continue by bounding the expected cost of the obtained solution.

**Claim 12.** *The algorithm opens at most $k + 3\lceil 2/(ab\eta) \rceil$ facilities.*

*Proof of Claim.* Recall that we have that $a\,|\mathcal{F}_1| + b\,|\mathcal{F}_2| = k$ and therefore

$$\sum_{i \in \mathcal{F}_1} (a + b\,|S_i|) = k. \tag{3}$$

First, consider a large star $i \in \mathcal{F}_1$, i.e., $a|S_i| \geq 1/(b\eta) \geq 1/\eta$. For such a star, the algorithm opens $1 + \lfloor b(|S_i| - 1) \rfloor \leq 1 + b(|S_i| - 1) = a + b|S_i|$ facilities, which is the contribution of star $i$ to (3).

Second, consider a group $\mathcal{U}_h$ of small stars and let $m := |\mathcal{U}_h|$. When considering this group, the algorithm opens $\lceil am \rceil + 1 \leq am + 2$ facilities in $\mathcal{F}_1$, and at most

$$(m - \lceil am \rceil - 1)h + \lceil bhm - (m - \lceil am \rceil - 1)h \rceil \leq bhm + 1$$

facilities in $\mathcal{F}_2$. Thus, the total number of facilities open from the group $\mathcal{U}_h$ of small stars is at most $m(a + bh) + 3$. As $m$ is the size of $\mathcal{U}_h$ and $a + bh$ is the contribution of each star in $\mathcal{U}_h$ to (3), the statement follows from that we have at most $\lceil 2/(ab\eta) \rceil$ groups. $\square$

We proceed by bounding the expected cost of the obtained solution. The intuition behind the following claim is that we have designed a randomized algorithm that opens a facility in $\mathcal{F}_2$ with probability $\approx b$ and a facility in $\mathcal{F}_1$ with probability $\approx a$. Therefore, if we connect a client $j$ to $i_2(j)$ with connection cost $d_2(j)$ if that facility is open, to $i_1(j)$ with connection cost $d_1(j)$ if that facility but not $i_2(j)$ is open, and to the center $i$ of the star $S_i : i_2(j) \in S_i$ with connection cost at most $2d_2(j) + d_1(j)$ if neither $i_1(j)$ or $i_2(j)$ are opened (recall that $i$ is open if not all facilities in $S_i$ are open), then the expected connection cost of client $j$ is at most

$$b \cdot d_2(j) + (1 - b)a \cdot d_1(j) + ab(2d_2(j) + d_1(j)) = ad_1(j) + b(1 + 2a)d_2(j).$$

The following claim then follows by linearity of expectation.

**Claim 13.** *The algorithm returns a solution with expected cost at most*

$$(1+\eta)\left(ad_1 + b(1+2a)d_2\right).$$

*Proof of Claim.* Focus on a client $j$ with $i_1(j) = i_1$ and $i_2(j) = i_2$ as depicted in Figure 3. Let $i_3 = \pi(i_2)$ be the closest facility in $\mathcal{F}_1$ to $i_2$, i.e., $i_3$ is the center of the star $S_{i_3}$ with $i_2 \in S_{i_3}$. Notice that $d(i_3, i_2) \le d(i_1, i_2) \le d_1(j) + d_2(j)$ by the definition of $\pi$. Thus, $d(j, i_3) \le d_2(j) + d(i_3, i_2) \le d_1(j) + 2d_2(j)$. We connect $j$ to $i_2$, if $i_2$ is open; otherwise, we connect $j$ to $i_1$ if $i_1$ is open. We connect $j$ to $i_3$ if both $i_1$ and $i_2$ are not open. (Notice that for a star $S_i$, if $i$ is not open, then all facilities in $S_i$ are open. Thus, either $i_2$ or $i_3$ is open.) Connecting $j$ to the nearest open facility can only give smaller connection cost. By abusing notations we let $i_1$ ($i_2$, resp.) denote the event that $i_1$ ($i_2$, resp.) is open and $\bar{i}_1$ ($\bar{i}_2$, resp.) denote the event that $i_1$ ($i_2$, resp.) is not open. Then, we can upper bound the expected connection cost of $j$ by

$$\Pr[i_2] \cdot d_2(j) + \Pr\left[i_1\bar{i}_2\right] \cdot d_1(j) + \Pr\left[\bar{i}_1\bar{i}_2\right] \cdot (2d_2(j) + d_1(j)),$$

which, by substituting $\Pr\left[\bar{i}_1\bar{i}_2\right] = 1 - \Pr[i_2] - \Pr\left[i_1\bar{i}_2\right]$, equals

$$\left(2 - \Pr[i_2] - 2\Pr\left[i_1\bar{i}_2\right]\right) d_2(j) + (1 - \Pr[i_2])\, d_1(j). \tag{4}$$

We upper bound this expression by analyzing these probabilities.

Let us start with $\Pr\left[i_1\bar{i}_2\right]$. If $i_2 \in S_{i_1}$ (i.e., $i_1 = i_3$) then $i_1$ is always open if $i_2$ is closed and thus we have $\Pr\left[i_1\bar{i}_2\right] = \Pr\left[\bar{i}_2\right]$. If $S_{i_1}$ is a large star, then $i_1$ is always open and we also have $\Pr\left[i_1\bar{i}_2\right] = \Pr\left[\bar{i}_2\right]$. In both cases, we have $\Pr\left[i_1\bar{i}_2\right] = 1 - \Pr[i_2]$.

We now consider the case where $S_{i_1}$ is a small star in a group $\mathcal{U}_h$ with $m := |\mathcal{U}_h|$ and $i_1 \neq i_3$. Note that if $S_{i_3}$ is either a large star or a small star not in $\mathcal{U}_h$ then the events $i_1$ and $\bar{i}_2$ are independent. We have thus in this case that

$$\begin{aligned}\Pr\left[i_1\bar{i}_2\right] &= \Pr[i_1] \cdot (1 - \Pr[i_2]) \\ &= \frac{\lceil am \rceil + 1}{m} \cdot (1 - \Pr[i_2])\end{aligned}$$

It remains to consider the case when $S_{i_3}$ is a star in $\mathcal{U}_h$. Notice that the dependence between $i_1$ and $i_2$ comes from that if $i_2$ is closed then $i_3$ is opened. Therefore, we have

$$\begin{aligned}\Pr\left[i_1\bar{i}_2\right] &= \Pr\left[i_1|\bar{i}_2\right] \cdot (1 - \Pr[i_2]) \\ &= \frac{\lceil am \rceil + 1 - 1}{m} \cdot (1 - \Pr[i_2]).\end{aligned}$$

We have thus showed that $\Pr\left[i_1\bar{i}_2\right]$ is always at least $a \cdot (1 - \Pr[i_2])$. Substituting in this bound in (4) allows us to upper bound the connection cost of $j$ by

$$(2b + (2a-1)\Pr[i_2])\, d_2(j) + (1 - \Pr[i_2])\, d_1(j).$$

We proceed by analyzing $\Pr[i_2]$. On the one hand, if $i_2$ is a leaf of some big star $S_i$ with $s = |S_i| \ge 2/(ba\eta)$ then $\Pr[i_2] = \frac{\lfloor b(s-1) \rfloor}{s}$ is greater than $b - 2/s \ge b(1 - a\eta)$ and smaller than $b$. On the other hand, if $i_2$ is a leaf of a small star $S_i$ in group $\mathcal{U}_h$ with $m := |\mathcal{U}_h|$ then in expectation we open exactly a $b$ fraction of the leaves so $\Pr[i_2] = b$. We have thus that $b(1 - a\eta) \le \Pr[i_2] \le b$. Since $(1+\eta) \cdot (1 - a\eta) \ge 1$ we have that the expected connection cost of facility $j$ is at most $(1+\eta)$ times

$$(2b + (2a-1)b)d_2(j) + (1-b)d_1(j) = b(1+2a)d_2(j) + ad_1(j).$$

The claim now follows by summing up the expected connection cost of all clients. $\square$

We complete the analysis by balancing the solution obtained by running our algorithm with the trivial solution of cost $d_1$ that opens all facilities in $\mathcal{F}_1$.

**Claim 14.** *We have that* $\min\{d_1, ad_1 + b(1+2a)d_2\} \le \frac{1+\sqrt{3}}{2}(ad_1 + bd_2)$.

*Proof of Claim.* We change $d_1$ and $d_2$ slightly so that $ad_1 + bd_2$ does not change. Apply the operation to the direction that increases the left-hand-side of the inequality. This operation can be applied until one of the 3 conditions is true: (1) $d_1 = 0$; (2) $d_2 = 0$ or (3) $d_1 = ad_1 + b(1+2a)d_2$.

For the first two cases, the inequality holds. In the third case, we have $d_1 = (1+2a)d_2$. Then $\frac{d_1}{ad_1+bd_2} = \frac{1+2a}{a(1+2a)+1-a} = \frac{1+2a}{1+2a^2}$. The maximum value of the quantity is $\frac{1+\sqrt{3}}{2}$, achieved when $a = \frac{\sqrt{3}-1}{2}$. □

We have shown that, by letting $\eta = \epsilon/(1+\sqrt{3})$, we can efficiently obtain a $O(1/\epsilon)$-additive $\frac{1+\sqrt{3}+\epsilon}{2}$-approximation to a bi-point solution with constant $a$ and $b$, which proves Theorem 5 when $a \in \left(\frac{\sqrt{3}-1}{4}, \frac{2}{1+\sqrt{3}}\right]$.

# 4 Discussion

We have given a $1+\sqrt{3}+\epsilon$-approximation algorithm for $k$-median, improving upon the previous best $3+\epsilon$-approximation algorithm. Besides the improved approximation guarantee, we believe that the most interesting technical contribution is Theorem 4, namely that we can approximate $k$ in $k$-median without loss of generality. More specifically, any pseudo-approximation algorithm which outputs a solution that opens $k + O(1)$ facilities can be turned into an approximation algorithm with essentially the same approximation guarantee but that only opens $k$ facilities.

For $k$-median this new point of view has the potential to overcome a known barrier for obtaining an approximation algorithm that matches the $1 + 2/e$ hardness of approximation result: the lower bound of 2 on the integrality gap of the natural LP for $k$-median. In particular, the known instances that give the integrality gap of 2 vanish if we allow $k + 1$ open facilities in the integral solution. Following our work, we therefore find it important to further understand the following open question: what is the maximum ratio between the cost of the optimum solution with $k+O(1)$ open facilities, and the value of the LP with $k$ open facilities? One can note that the hardness of approximation reduction in [11] implies that the integrality gap is at least $1 + 2/e$ even if we open $k + o(k)$ facilities. Moreover our $O(1/\epsilon)$-additive approximation for bi-point solutions achieving a guarantee of $\frac{1+\sqrt{3}+\epsilon}{2} < 1 + 2/e$ shows that the worst case integrality gap instances are not of this type when pseudo-approximation is allowed.

Finally, we would like to mention that Theorem 4 naturally motivates the question if other hard constraints can be relaxed to soft constraints with a "violation-dependent" increase in the runtime. Soft constraints often greatly help when designing algorithms. For example, the capacitated versions of facility location and $k$-median are notorious problems when the capacities are hard constraints but better approximation algorithms are known if the capacities are allowed to be slightly violated (see e.g. [8]). As our approach was inspired by studying the power of the Sherali-Adams hierarchy [21] for the $k$-median problem, we believe that a promising research direction is to understand the power of that hierarchy and the stronger Lasserre hierarchy [15] when applied to these kinds of problems.

# References


[1] A. Archer, R. Rajagopalan, and D. B. Shmoys. Lagrangian relaxation for the k-median problem: new insights and continuity properties. In *In Proceedings of the 11th Annual European Symposium on Algorithms*, pages 31–42, 2003.

[2] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristic for k-median and facility location problems. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, STOC '01, pages 21–29, New York, NY, USA, 2001. ACM.

[3] P. S. Bradley, Fayyad U. M., and O. L. Mangasarian. Mathematical programming for data mining: Formulations and challenges. *INFORMS Journal on Computing*, 11:217–238, 1998.

[4] J. Byrka. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. In *APPROX '07/RANDOM '07: Proceedings of the 10th International Workshop on Approximation and the 11th International Workshop on Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 29–43, Berlin, Heidelberg, 2007. Springer-Verlag.

[5] M. Charikar and S. Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *In Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 378–388, 1999.

[6] M. Charikar, S. Guha, E. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the k-median problem (extended abstract). In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, STOC '99, pages 1–10, New York, NY, USA, 1999. ACM.

[7] F. A. Chudak and D. B. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM J. Comput.*, 33(1):1–25, 2004.

[8] J. Chuzhoy and Y. Rabani. Approximating k-median with non-uniform capacities. In *SODA*, pages 952–958, 2005.

[9] S Guha and S Khuller. Greedy strikes back: Improved facility location algorithms. In *Journal of Algorithms*, pages 649–657, 1998.

[10] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *J. ACM*, 50:795–824, November 2003.

[11] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, STOC '02, pages 731–740, New York, NY, USA, 2002. ACM.

[12] K Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.

[13] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. In *Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, SODA '98, pages 1–10, Philadelphia, PA, USA, 1998. Society for Industrial and Applied Mathematics.

[14] A. A. Kuehn and M. J. Hamburger. A heuristic program for locating warehouses. 9(9):643–666, July 1963.

[15] J. B. Lasserre. An explicit equivalent positive semidefinite program for nonlinear 0-1 programs. *SIAM Journal on Optimization*, 12(3):756–769, 2002.

[16] S. Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. In *Automata, Languages and Programming - 38th International Colloquium (ICALP)*, pages 77–88, 2011.

[17] J. Lin and J. S. Vitter. Approximation algorithms for geometric median problems. *Inf. Process. Lett.*, 44:245–249, December 1992.

[18] J. Lin and J. S. Vitter. $\epsilon$-approximations with minimum packing constraint violation (extended abstract). In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC), Victoria, British Columbia, Canada*, pages 771–782, 1992.

[19] M. Mahdian, Y. Ye, and J. Zhang. Approximation algorithms for metric facility location problems. *SIAM J. Comput.*, 36(2):411–432, 2006.

[20] A. S. Manne. Plant location under economies-of-scale-decentralization and computation. In *Managment Science*, 1964.

[21] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.*, 3(3):411–430, 1990.

[22] D. B. Shmoys, E. Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract). In *STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 265–274, New York, NY, USA, 1997. ACM.